**Мини-отчет**

*Группа №2*

Так как в основе нашей методики лежит обучение без учителя, у нас не было необходимости ограничивать объём development-корпуса. Тем не менее, для определения характерных слов с негативной и позитивной окраской мы выбрали все тексты с оценками от 0 до 2 включительно за еду (2132 текста) и сервис (2731 текст), и аналогичным образом с оценками от 8 до 10 включительно за еду (11152 текста) и сервис (10916 текстов). Необходимости дополнительно балансировать корпус не было, так как для нас определение позитивных и негативных отзывов - фактически, независимые задачи, и мы приняли меры для дополнительного разграничения этих классов: в таких условиях искусственно ограничивать объём корпуса было не очень осмысленно. Мы не работали с иной метаинформацией, помимо оценок, но мы использовали в качестве исходных данных (seed words) предложенные наборы слов про еду и сервис, которые мы вручную дополнили. Дополнения касались в первую очередь отрицательных слов, которых было достаточно мало. Мы дополнили эти отрицательные наборы очень общими словами с отрицательной оценкой, вроде “плохой” и т.д., а также несколькими тематическими вроде “протухший”, “грубый” и т.д. В итоге для каждого типа текстов (позитивные/негативные; о еде/о сервисе) у нас получился набор примерно из 8-14 seedwords.

Основной метод, которые мы применили, заключается в автоматическом определении широкого класса релевантных слов и их дальнейшему нахождению в текстах по словарю. Для этого мы использовали на первой стадии метрику странности слов, а затем — векторной близости к найденным словам.

Под странностью мы считали в данном случае соотношение log(pspec(wi)) / log(pcomp(wi)), где wi - каждое слово в “полярном тексте”, p\_spec - частота этого слова в текстах этой же полярности (этого же аспекта), p\_comp - частота этого слова в противоположной полярности (этого же аспекта). После вычисления странностей для каждого слова (исключая стоп-слова) и биграммы (включая стоп-слова и пунктуацию), мы для каждого из четырех типов текстов отранжировали их в убывающем порядке (от самых “странных” к самым “общим”).

Далее мы использовали вектора. В качестве векторной модели мы использовали Tayga UPOS Skipgram (объем - 5 млрд слов, вес - 610 Мбайт). Поиск слов в тренировочных текстах мы устроили так:

1. берем вектора для всех seedwords внутри каждой группы и усредняем их.

2. ищем для каждого из слов и биграмм в нашем “списке странности” вектор в модели и сравниваем его с усредненным вектором. Если его косинусная близость больше 0.5 И косинусная близость с вектором меньше как минимум на 0.2, мы добавляем его в наш лексикон.

2а. Также для первых десяти попавших в лексикон слов мы добавляем их в наш усредненный вектор. (т.е. высчитываем среднее между усредненным вектором и вектором нового слова).

3. Цикл 1-2 мы повторяем дважды для каждой группы слов.

Таким образом, мы получили лексиконы для каждой из четырех групп текстов - 116 токенов (слов + биграмм) для Negative Food, 41 токен для Negative Service и более чем по тысяче токенов для Positive Food и Positive Service. В финальном документе lexicon.txt мы собрали все униграммы - получилось 163 лексемы. Также в папке lexicon\_by\_parts лежат (разделенные по оценкам и темам) униграммы и биграммы (которых суммарно получилось более 3 тысяч). Все они используются при предикшне.

При предикшне мы также используем два способа вычленения эмоционально окрашенных слов: сначала мы ищем каждое из слов из тестового набора в списках наших лексиконов; затем мы итеративно определяем вектор для каждого слова и находим векторную близость между словом и усредненным вектором для каждой из наших четырёх категорий — это позволяет нам обойти проблему OOV.